Quantum Information and Computation, Vol. 0, No. 0 (2010) 000–000

INFORMATION RECONCILIATION FOR QUANTUM KEY DISTRIBUTION

DAVID ELKOUSS, JESUS MARTINEZ-MATEO, VICENTE MARTIN[a]
Research group on Quantum Information and Computation[b]
Facultad de Informática, Universidad Politécnica de Madrid
Campus de Montegancedo, 28660 Boadilla del Monte, Madrid, Spain

Received (received date)
Revised (revised date)

Abstract

Quantum key distribution (QKD) relies on quantum and classical procedures in order to achieve the growing of a secret random string —the key— known only to the two parties executing the protocol. Limited intrinsic efficiency of the protocol, imperfect devices and eavesdropping produce errors and information leakage from which the set of measured signals —the raw key— must be stripped in order to distill a final, information theoretically secure, key. The key distillation process is a classical one in which basis reconciliation, error correction and privacy amplification protocols are applied to the raw key. This cleaning process is known as information reconciliation and must be done in a fast and efficient way to avoid cramping the performance of the QKD system. Brassard and Salvail proposed a very simple and elegant protocol to reconcile keys in the secret-key agreement context, known as *Cascade*, that has become the de-facto standard for all QKD practical implementations. However, it is highly interactive, requiring many communications between the legitimate parties and its efficiency is not optimal, imposing an early limit to the maximum tolerable error rate. In this paper we describe a low-density parity-check reconciliation protocol that improves significantly on these problems. The protocol exhibits better efficiency and limits the number of uses of the communications channel. It is also able to adapt to different error rates while remaining efficient, thus reaching longer distances or higher secure key rate for a given QKD system.

Keywords: Quantum cryptography, quantum key distribution, information reconciliation, rate-compatible, low-density parity-check codes
Communicated by: to be filled by the Editorial

1 Introduction

A quantum key distribution protocol is composed of two parts: a quantum and a classical one [1]. The quantum part involves the actual transmission of qubits, its manipulation and detection, and it is performed using a quantum channel. The classical part is done through a public, albeit integrity-preserving, classical channel and involves basis reconciliation, error correction and privacy amplification protocols. The quantum part results

[a]vicente@fi.upm.es
[b]http://gcc.ls.fi.upm.es

in the production of a raw key at both ends of the quantum channel. The raw key must be cleaned from all the unavoidable errors produced in this part. These include the intrinsic ones, due to the limited efficiency of the protocol, and those arising either from the inevitable imperfections in the physical setup or from eavesdropping. Intrinsic errors are easier to correct and this is usually done by bookkeeping of the detection events and subsequent discussion over the classical channel about the preparation state of the qubits leading to the recorded events. For example, in the standard BB84 protocol [2], half of the qubits detected by Bob will be in a base orthogonal to the one in which they were prepared by Alice, leading to a 50% of detections that do not directly contribute bits to the final secret key. In a real setup, the remaining bits will still be affected from errors arising either from the physical implementation itself or from an eavesdropper, these being in principle indistinguishable. The process to clean the key from the errors is known as information reconciliation and is done through the classical channel.

The existence of optimal, although inefficient, protocols leaking a minimum of information in the process was demonstrated in [3]. There, a practical protocol trading an acceptable amount of leaked information for efficiency was also proposed. This reconciliation protocol, known as *Cascade*, has become the de-facto standard for all QKD practical implementations. However, it has several shortcomings that make it less than ideal under certain situations that are expected to become more common in real world environments. Work has been done to improve on *Cascade* [4–7], but none of the resulting methods have become as widespread.

Recent advances in QKD systems have seen a tremendous increase in key generation speed [8–10]. Current generation systems can be successfully used over longer distances or in noisier environments than before, like those arising when integration with conventional networks is required [11,12]. This changes indicate that the reconciliation protocol must be efficient at high key and error rates.

Cascade is a highly interactive protocol that requires a high number of uses of the public channel to proceed. The number of uses raises markedly with the quantum bit error rate (QBER), thus it is not well suited for next generation QKD systems. From a practical point of view, the protocol must be implemented using two computers at both ends of the quantum channel that process the key and communicate through the public channel. The typical access latencies to the network are much higher than CPU operations, hence it is easy to produce a bottleneck in highly interactive protocols. In fact, if a specialised communications network is not used, the communications needs of *Cascade* are already the limiting factor in many situations and with current systems, instead of the much more delicate quantum part. On the other hand, a less than ideal efficiency limits the maximum number of errors that can be corrected without publishing too much information, thus reducing the performance of the system when working at a high error rate.

In this paper we describe a reconciliation protocol that overcomes these problems. The protocol exhibits a better efficiency than *Cascade*, it can be adapted to a varying QBER with a low information leakage, extending the usable range of the system. It limits the number of uses of the public channel and its structure allows for a hardware implementation, avoiding communications and CPU bottlenecks, thus being well suited

to next generation QKD systems in demanding environments. LDPC codes also have an structure that make them well suited for hardware implementations

The paper is organised as follows: In Section 2, the information reconciliation problem in the secret-key agreement context is described and the current status of error correction in QKD is discussed. A new Information Reconciliation Protocol able to adapt to different channel parameters is presented and its asymptotic behavior discussed in Section 3. In Section 4 the results of a practical implementation of the protocol are shown. In particular the efficiency of the protocol is compared to its optimal theoretical value and to *Cascade*.

2 Problem statement

2.1 Information reconciliation

Let Alice and Bob be two parties with access to dependent sources identified by two random variables, X and Y respectively. Information reconciliation is the process by which Alice and Bob extract common information from their correlated sources. In a practical setting Alice and Bob hold $\mathbf{x}$ and $\mathbf{y}$, two n-length strings that are the outcome of X and Y, and they will agree in some string $\mathbf{s} = f(\mathbf{x}, \mathbf{y})$ through one-way or bidirectional conversation [13]. The conversation $\phi(\mathbf{x}, \mathbf{y})$ is also a function of the outcome strings, and its quality can be measured by the number of symbols involved in the conversation $M = |\phi(\mathbf{x}, \mathbf{y})|$.

Now, the problem of encoding correlated sources is a well known problem in information theory. To independently encode X and Y at least a rate $R \geq H(X) + H(Y)$ is needed. However, in their seminal paper, Slepian and Wolf [14] demonstrated that to jointly encode both variables it is enough with a rate $R \geq H(X, Y)$ even if X and Y are encoded separately. Moreover, if Y is available at the decoder only a rate of $R \geq H(X|Y)$ is needed to encode X (see Fig. 1), which in the information reconciliation context amounts for the minimum information needed in order to reconcile Alice's and Bob's strings. To measure the quality of a real reconciliation schema, that in a practical setting will encode X with a higher rate than $H(X|Y)$, we use the efficiency parameter $f \geq 1$ defined as:

$$I_{\text{real}} = fH(X|Y) \geq I_{\text{opt}} \tag{1}$$

where I_{real} is the information published during the reconciliation process, I_{opt} is the minimum information that would allow to reconcile Alice's and Bob's strings, and $H(X|Y)$ is the conditional Shannon entropy.

However, there are two other parameters to consider when evaluating the quality of a information reconciliation procedure: that is the computational complexity and the interactivity. The first one stresses that a real information reconciliation procedure must be feasible. Any sufficiently long random linear code of the appropriate rate could solve the problem [15], however optimal decoding is in general an NP-hard problem. The interactivity of a reconciliation protocol has to be taken into account because, specially in high latency scenarios, the communications overhead can pose a severe burden on the performance of the QKD protocol.

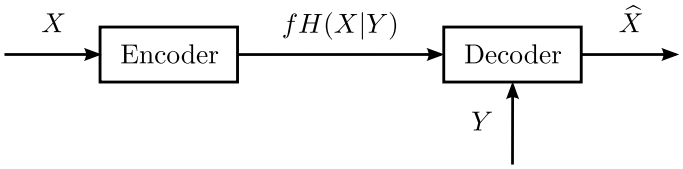

Fig. 1. Source coding with side information.

In order to evaluate the quality of the reconciliation, we will concentrate in discrete variable QKD protocols even if the ideas presented here can be easily extrapolated to other scenarios. Most QKD protocols encode the information in discrete variables [2,16], although there are many proposals on continuous variable protocols [17–19]. Errors on the quantum channel are normally uncorrelated and symmetric or, if prior to the reconciliation Alice and Bob apply a random permutation, they can behave as such [20]. In this situation Alice's and Bob's strings can be regarded as the input and output of a binary symmetric channel (BSC), characterized by the crossover probability ε, and the efficiency parameter f can be described as the relationship between the length of the conversation $M = |\phi(\mathbf{x}, \mathbf{y})|$ and the optimal value $N \cdot H(X|Y) = N \cdot h(\varepsilon)$:

$$f = \frac{M}{N \cdot h(\varepsilon)} \tag{2}$$

It was first shown in [21] that low-density parity-check (LDPC) codes used within Wyner's coset scheme [15, 22] are a good solution for the compression of binary sources with side information. LDPC codes [26] are linear codes that have a sparse parity check matrix. These codes, when decoded with the belief propagation algorithm, can perform very close to the theoretical limit.

The fundamental idea is to assign each source vector to a bin from a set of $2^{H(X|Y)+\iota}$ known bins, the encoder describes the bin to the decoder, and the decoder searches for the source vector inside the described bin. Let $\mathbf{x}$ and $\mathbf{y}$ be two binary strings of length n, and C a $[n, k]$ binary linear code specified by its parity matrix H. The syndrome $\mathbf{s}$ of a vector $\mathbf{x}$ is the $n - k$ string defined as $\mathbf{s} = H\mathbf{x}$ being $\mathbf{s} = \mathbf{0}$ for all the codewords in C. Each syndrome $\mathbf{s}$ defines a coset $C_\mathbf{s}$ as the set of all strings, $\{\mathbf{x}\}$, that verify $H\mathbf{x} = \mathbf{s}$. Wyner's schema consists in assigning each source vector to one of the cosets of C. Encoding $\mathbf{x}$ amounts then to compute its syndrome, and decoding is simply to find the member in $C_\mathbf{s}$ closest to $\mathbf{y}$. An LDPC message passing decoder was modified in [21] to take into account the syndrome decoding proposed by Wyner. This same procedure can be applied to the QKD scenario.

2.2 Previous work

As mentioned in the introduction, the most widely used and best known protocol for error correction in the QKD context is *Cascade*. Proposed by Brassard and Salvail [3], this protocol runs for a fixed number of passes. In each pass, Alice and Bob divide their strings into blocks of equal length. The initial block length depends on the estimated error probability, p, and it is doubled when starting a new pass. For each block they

compute and exchange its parity. A parity mismatch implies an odd number of errors, and a dichotomic search allows both parties to find one of the errors. Whenever an error is found after the first pass, it uncovers an odd number of errors masked on the preceding passes and the algorithm returns to correct those errors previously undetected. This cascading process gives name to the protocol. Several papers propose improvements on *Cascade* [4, 5], these papers analyse how the block length is to be chosen and increased in order to optimise the efficiency of the reconciliation, but the main characteristics remain unaltered. *Cascade*, although highly interactive, is reasonably efficient and easy to implement. It is well known and has become the de-facto standard, hence we have chosen *Cascade* as the benchmark to compare against.

Winnow [6] is another well know reconciliation protocol in the QKD context, it requires only two communications between the parties. In the first communication Alice and Bob exchange the parities of every block. After that, they exchange the syndrome of a Hamming code for correcting single errors in each block with a parity mismatch. The protocol incorporates a privacy maintenance procedure by discarding one bit per parity revealed (i.e. m bits are discarded when a syndrome of length m is exchanged). Its main advantage is a reduction on the number of communication needed, however the efficiency of the protocol is worse than that of *Cascade* in the error range of interest. Recently, some interesting improvements have been proposed for selecting an optimum block length in this protocol [7].

Modern coding techniques have not been applied to discrete variable QKD until recently. LDPC codes were proposed and used on [23]. But as the codes had not been specifically designed for the problem, aside from the inherent advantage of forward error correction, the efficiency was worse than that of *Cascade*. These codes have been also used in the context of continuous variable QKD [19]. LDPC codes were first optimized for the BSC on [24], and although the results were close to optimal for the designed codes, the efficiency curve exhibited a saw behaviour due to a lack of information rate adaptability in the proposed procedure [27]. Since the error rate can vary among transmissions, it is important for a protocol to be able to cope with this change.

3 Rate-compatible reconciliation

Although linear codes are a good solution for the reconciliation problem, since they can be tailored to a given error rate, their efficiency degrades when it is not known beforehand. This is the case in QKD, where the error rate is an a priori unknown that is estimated for every exchange. The QBER might vary significantly in two consecutive key exchanges, specially when the quantum channel is transported through a shared optical fibre that can be used together with several independent classical or quantum channels that can add noise. To address this problem there are two different options: (i) it is possible to build a code once the error rate has been estimated, and (ii) a pre-built code can be modified to adjust its information rate. The computational overhead would make the first option almost unfeasible except for very stable quantum channels, something difficult to achieve in practise and impossible in the case of a shared quantum channel in a reconfigurable network environment [11]. In this paper we propose the use of the second strategy as the easiest and most effective way to obtain a code for the required

rate, for which we describe a protocol that adapts pre-built codes in real time while maintaining an efficiency close to the optimal value.

3.1 Rate modulation

Puncturing and shortening are two common strategies used to adapt the rate of a linear code. This process of adapting the information rate of a pre-built code will be referred as rate modulation. When p punctured symbols of a codeword are removed, a $[n, k]$ code is converted into a $[n-p, k]$ code. Whereas, when shortening, s symbols are removed during the encoding process, and a $[n, k]$ code is converted into a $[n-s, k-s]$ code. A graphical representation, on a Tanner graph, of the procedures just described for puncturing and shortening and its effects on the rate of the sample code is shown in Fig. 2.

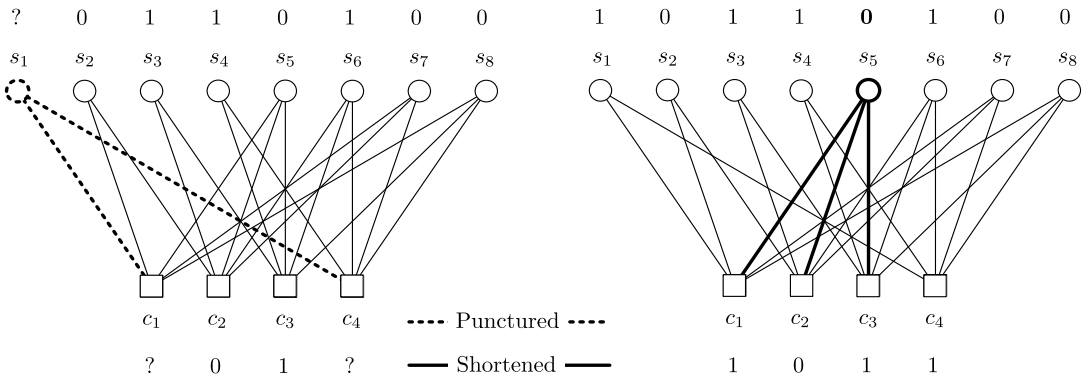

Fig. 2. Examples of puncturing and shortening strategies applied to a linear code represented by its Tanner graph. In the puncturing example (left) one symbol is deleted from the word and a [8,4] code, with rate $R = 1/2$, is converted to a [7,4], increasing its rate to $R = 4/7$. In the shortening example (right), one symbol is deleted from the encoding and the same [8,4] code is converted to a [7,3] code, the rate now decreases to $R = 3/7$.

These procedures may be regarded as the transmission of different parts of the codeword over different channels (see Fig. 3). Since puncturing is a process by which p codeword symbols are eliminated, it can be seen as a transmission over a binary erasure channel (BEC) with erasure probability of 1, BEC(1). Shortening is a process by which s codeword symbols are known with absolute certainty, as such it can be seen as a transmission over a BEC with erasure probability of 0, BEC(0). The remaining symbols are transmitted by the real channel which in the present paper can be modelled by a binary symmetric channel with crossover probability ε, BSC(ε)

Supposing that R_0 is the original coding rate, the modulated rate is then calculated as:

$$R = \frac{R_0 - \sigma}{1 - \pi - \sigma} = \frac{k - s}{n - p - s} \tag{3}$$

where π and σ represent the ratios of information punctured and shortened respectively.

Both strategies, puncturing and shortening, can be applied simultaneously. Given a $[n, k]$ code and $n' \leq n$ bits, if puncturing and shortening are applied with a constant

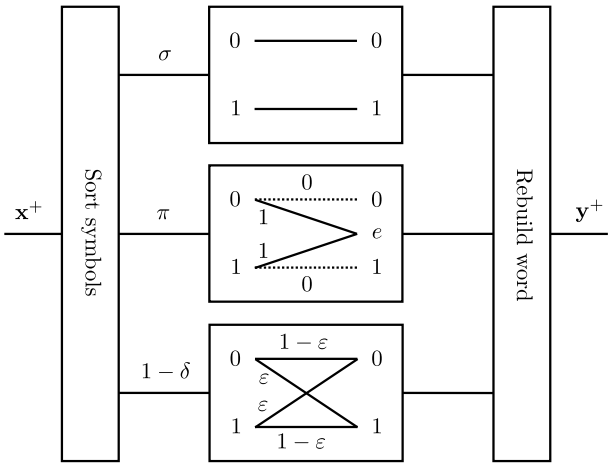

Fig. 3. Channel model. Puncturing and shortening on a LDPC code results in the division of the original binary symmetric channel used to reconcile Alice's $\mathbf{x}$ string with Bob's $\mathbf{y}$ into three different channels: a binary erasure channel with erasure probability of 1 (for the fraction π of punctured symbols), a BEC with erasure probability of 0 (for the fraction σ of shortened symbols) and a binary symmetric channel with crossover probability ε (for the rest of the symbols).

number d of punctured and shortened symbols, a single code can be used to protect the n' bits for different error rates. There are two consequences of applying a constant d: (i) there is a limit to the minimum and maximum achievable information rates. These limits, expressed as a function of $\delta = d/n$, define the correction interval:

$$0 \leq R_{\min} = \frac{R_0 - \delta}{1-\delta} \leq R \leq \frac{R_0}{1-\delta} = R_{\max} \leq 1 \tag{4}$$

(ii) puncturing and shortening procedures cause an efficiency loss [28]. Therefore, there is a tradeoff between the achievable information rates and reconciliation efficiency.

This efficiency loss, caused by high levels of puncturing and shortening, can be avoided if a set of n codes ζ_i with different information rates is used: $R_0(\zeta_1) \leq R_0(\zeta_2) \leq R_0(\zeta_n)$. The target error range can then be partitioned into, $[R_{\min}(\zeta_1), R_{\max}(\zeta_1)] \cup [R_{\min}(\zeta_2), R_{\max}(\zeta_2)] \cup ... \cup [R_{\min}(\zeta_n), R_{\max}(\zeta_n)]$, not necessarily with the same size. The number of intervals depends on the width of the error rate range to cover and on the desired efficiency. The compromise between the width of the interval covered and the achieved efficiency in the one code case is transferred to a compromise between efficiency and the added complexity of managing several codes. Fig. 4 shows the computed efficiency thresholds for several families of codes with different coding rates.

3.2 Protocol

We now proceed to describe a rate-compatible information reconciliation protocol using puncturing and shortening techniques as described above.

Step 0: Raw key exchange. Alice and Bob obtain a raw key by running a QKD protocol through a quantum channel (see Section 2). This key exchange may be modelled

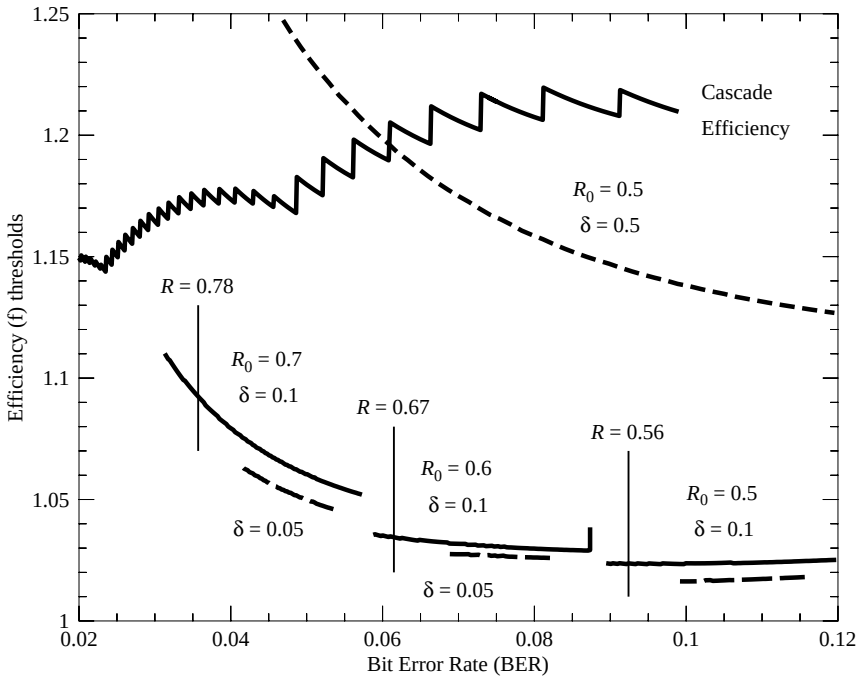

Fig. 4. Efficiency thresholds for different codes with information rates, $R_0 = 0.5$, 0.6 and 0.7 as a function of the quantum bit error rate (QBER). Two δ values, 0.1 (solid line) and 0.05 (dashed) have been used to adapt the rate for each code. As a comparison, a single code covering all of the QBER range of interest, with rate $R_0 = 0.5$ and $\delta = 0.5$, is presented to show how the efficiency degrades for high δ values, although a broader range is covered. The codes have been optimised using the density evolution algorithm for the BSC. The *Cascade* efficiency was calculated using the same sample size (2×10^5). The block size used in the first step, k_1, is given by $k_1 = \lceil 0.73/QBER \rceil$ (optimized in [33]) and doubled in every subsequent step $k_n = 2k_{n-1}$. The sawtooth behaviour of the *Cascade* efficiency reflects the points where k_1 changes.

as follows. Alice sends to Bob the string $\mathbf{x}$, an instance of a random variable X, of length $\ell = n - d$ through a binary symmetric channel with crossover probability ε, BSC(ε) (or a black box behaving as such). Bob receives the correlated string, $\mathbf{y}$, but with discrepancies to be removed in the following steps.

Step 1: Pre-conditions. Prior to the key reconciliation process Alice and Bob agree on the following parameters: (i) a pool of shared codes of length n, constructed for different coding rates; (ii) the size of the sample, t, that will be used to estimate the error rate in the communication; and (iii) the maximum number of symbols that will be punctured or shortened to adapt the coding rate, $d = p + s = n\delta$.

Step 2: Error rate estimation. Bob chooses randomly a sample of t bits of $\mathbf{y}$, $\alpha(\mathbf{y})$, and sends them and their positions, $\beta(\mathbf{y})$, to Alice through a noiseless channel (i.e. the public and integrity-preserving channel used in the classic part of a QKD protocol). Using the positions received from Bob, $\beta(\mathbf{y})$, Alice extracts an equivalent sample in $\mathbf{x}$, $\alpha(\mathbf{x})$, and estimates the crossover probability for the exchanged key by comparing the two samples:

$$\varepsilon' = \frac{\alpha(\mathbf{x}) + \alpha(\mathbf{y})}{t} \tag{5}$$

Once Alice has estimated ε', she knows the theoretical rate for a punctured and shortened code able to correct the string. Now she computes the optimal rate corresponding to the efficiency of the code she is using: $R = 1 - f(\varepsilon')h(\varepsilon')$; where h is the binary Shannon entropy function and f the efficiency. Then she can derive the optimal values for puncturing and shortening, p and s respectively, as:

$$\begin{aligned} s &= \lceil (R_0 - R(1 - d/n)) \cdot n \rceil \\ p &= d - s \end{aligned} \tag{6}$$

Step 3: Coding. Alice creates a string $\mathbf{x}^+ = g(\mathbf{x}, \sigma_{\varepsilon'}, \pi_{\varepsilon'})$ of size n. The function g defines the $n - d$ positions to take the values of string $\mathbf{x}$, the p positions to be assigned random values, and the s positions to have values known by Alice and Bob. The set of $n - d$ positions, the set of p positions and the set of s positions and their values come from a synchronised pseudo-random generator. She then sends $s(\mathbf{x}^+)$, the syndrome of $\mathbf{x}^+$, to Bob as well as the estimated crossover probability ε'.

This process can be regarded as jointly coding (and decoding) the original strings sent through a BSC(QBER) with p bits sent through a binary erasure channel (BEC) with erasure probability 1, and s bits sent through a noiseless channel (see Fig. 3).

Step 4: Decoding. Bob can reproduce Alice's estimation of the optimal rate R, the positions of the p punctured bits, and the positions and values of the s shortened bits. Bob then creates the corresponding string $\mathbf{y}^+ = g(\mathbf{y}, \sigma_{\varepsilon'}, \pi_{\varepsilon'})$. He should now be able to decode Alice's codeword with high probability, as the rate has been adapted to the channel crossover probability. Bob sends an acknowledgement to Alice to indicate if he successfully recovered $\mathbf{x}^+$.

Step 5: (Optional) Interactive decoding. If Bob does not succeed in recovering Alice's string, Alice can reduce the information rate of the code by revealing some $r^* \leq p$ of the punctured bits on the public channel, that become shortened bits (see Fig. 5). Steps 3

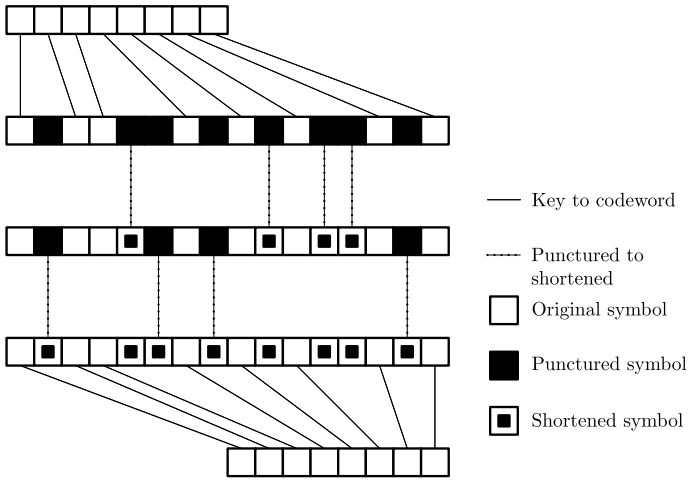

Fig. 5. Protocol sequence for interactive reconciliation. The figure shows how punctured symbols are converted into shortened symbols in each round, thus repeating coding and decoding steps for different coding rates. The interactive protocol concludes when the target string has been reconciled or there are no more punctured symbols to be revealed.

and 4 are then repeated, Alice computes the new syndrome and sends it to Bob, who tries to decode and send an acknowledge to Alice. Let $p^{(i)}$ and $s^{(i)}$ be the number of punctured and shortened bits respectively in the i-th round of the proposed protocol, and $r^{(i+1)}$ the number of punctured bits to be revealed for the next round, the new proportion of punctured and shortened bits used for the reconciliation are calculated as $p^{(i+1)} = p^{(i)} - r^{(i+1)}$ and $s^{(i+1)} = s^{(i)} + r^{(i+1)}$, respectively. These steps can be repeated while Bob does not find the correct string and there are punctured bits that have not been revealed as shortened bits, i.e. while $p \geq 0$.

4 Simulation results

In this section we discuss the efficiency of the rate-compatible information reconciliation protocol without the interactive decoding step, comparing the results of the protocol to regular LDPC codes as proposed in [24] and to *Cascade*. The purpose of this simulations is to highlight that the proposed protocol extends the working range in QKD, allowing to distill a key in a wider QBER range than previous information reconciliation protocols.

Fig. 6 shows the efficiency, calculated as defined in Eq. (1), in the reconciliation process simulated for three different alternatives: (i) using the *Cascade* protocol, (ii) using LDPC codes without adapting the information rate, and (iii) using LDPC codes adapting the information rate with the rate-compatible protocol proposed here. The target error range selected is $[0.055, 0.11]$, where a high efficiency protocol is a must. Low QBER rates do not demand a close to optimal efficiency since other requisites, such as the throughput, are more critical in obtaining a high secret key rate. In order to achieve a efficiency close to 1, the error range $[0.055, 0.11]$ has been divided into two correction intervals: $R_0(\zeta_1) = 0.5$, $R_0(\zeta_2) = 0.6$ and $\delta = 0.1$. The codes have been constructed

using families of LDPC codes specifically optimised for the BSC. Generating polynomials of these families can be found in [24], however they were not designed for shortening and puncturing. Taking into account these parameters in the generating polynomial design process would allow to cover the whole QBER range with high efficiency.

The construction process has been also optimised using a modified progressive edge-growth algorithm for irregular codes with a detailed check node degree distribution [32]. A codeword length of 2×10^5 bits has been used.

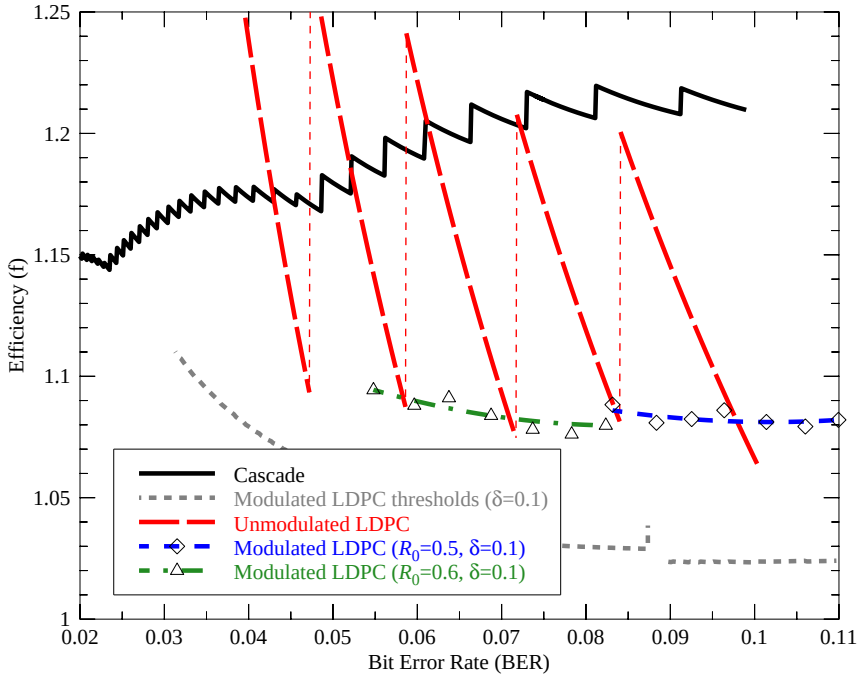

Fig. 6. Computed efficiency for medium to high error rates, a typical range expected in shared quantum channel environments, long distances or high losses scenarios, such as in networks, and where obtaining high efficiency is critical. The solid line is the *Cascade* efficiency. Its parameters are the same than for Fig. 4. The dotted line represents the modulated LDPC thresholds. For all LDPC results shown here $\delta = 0.1$. The long, thick, dashed lines joined by thin dashed lines is the efficiency of an unmodulated code. Short dash and dash-dotted lines are the results for the modulated codes. Dash-dotted is for a rate $R_0 = 0.6$ and short dash are for $R_0 = 0.5$, triangles and diamonds are used to mark the computed points. The smooth and efficient behaviour of the modulated, rate adapted codes, as compared to the unmodulated version is to be noted. The gain in efficiency over *Cascade* allows for an extended usability range of the system at high QBER.

The results show that there is a small price to pay for the rate adaptation. LDPC codes without puncturing and shortening behave slightly better near their threshold, however for the δ value chosen the penalty is very small and the rate-compatible protocol allows to reconcile strings in all the range with $f \leq 1.1$. The unmodulated LDPC codes exhibit an undesirable saw behaviour that can lead to efficiencies worse than that of *Cascade* unless many different codes are calculated, incurring in an unacceptable penalty

in CPU time. The new protocol works at a much better efficiency than *Cascade*, that performs in all the tested range with $f \geq 1.17$.

5 Conclusions

We have demonstrated how to adapt an LDPC code for rate compatibility. The capability to adapt to different error rates while minimizing the amount of published information is an important feature for secret-key reconciliation in the QKD context, specially whenever it is used in long distance links or in noisy environments such as those arising in shared optical networks. In these demanding environments high efficiency is necessary to distill a key. The protocol improves on *Cascade*, allowing to reach efficiencies close to one while limiting the information leakage and having the important practical advantage of low interactivity: only one message is exchanged by both parties. This high efficiency allows to extend the working range in QKD, that is, it allows to distill a key in a wider QBER range than previous information reconciliation protocols.

Acknowledgment

This work has been partially supported by the project Quantum Information Technologies in Madrid[c](QUITEMAD), Project P2009/ESP-1594, *Comunidad Autónoma de Madrid*.

The authors gratefully acknowledge the computer resources, technical expertise and assistance provided by the *Centro de Supercomputación y Visualización de Madrid*[d](CeSViMa) and the Spanish Supercomputing Network.

References

1. N. Gisin, G. Ribordy, W. Tittel and H. Zbinden (2002), *Quantum cryptography*, Rev. Mod. Phys., Vol. 74, pp. 145-195.
2. C.H. Bennett and G. Brassard (1984), *Quantum cryptography: public key distribution and coin tossing*, in Proceedings of IEEE International Conference on Computers, Systems and Signal Processing, IEEE press., pp. 175-179.
3. G. Brassard and L. Salvail (1994), *Secret-Key Reconciliation by Public Discussion*, in Eurocrypt'93: Workshop on the theory and application of cryptographic techniques on Advances in cryptology, Lecture Notes in Computer Science, Vol. 765, pp. 410-423.
4. T. Sugimoto and K. Yamazaki (2000), *A study on secret key reconciliation protocol "Cascade"*, IEICE Trans. Fundam. Electron. Commun. Comput. Sci., Vol. E83-A, No. 10, pp. 1987-1991.
5. S. Liu, H.C.A. Van Tilborg and M. Van Dijk (2003), *A Practical Protocol for Advantage Distillation and Information Reconciliation*, Designs Codes Cryptogr., Vol. 30, No. 1, pp. 39-62.
6. W.T. Buttler, S.K. Lamoreaux, J.R. Torgerson, G.H. Nickel and C.G. Peterson (2003), *Fast, efficient error reconciliation for quantum cryptography* Phys. Rev. A, Vol. 67, No. 5, p. 052303.
7. J. Han, and X. Qian (2009), *Auto-adaptive interval selection for quantum key distribution*, Quantum Inform. Comput., Vol. 9, No. 7&8, pp. 693-700.

[c]http://www.quitemad.org
[d]http://www.cesvima.upm.es

8. A.R. Dixon, Z.L. Yuan, J.F. Dynes, A.W. Sharpe and A.J. Shields (2008), *Gigahertz decoy quantum key distribution with 1 Mbit/s secure key rate*, Opt. Express, Vol. 16, No. 23, pp. 18790-18979.
9. D. Stucki, N. Brunner, N. Gisin, V. Scarani and H. Zbinden (2005), *Fast and simple one-way quantum key distribution*, Appl. Phys. Lett., Vol. 87, No. 19, p. 194108.
10. K.J. Gordon, V. Fernandez, G.S. Buller, I. Rech, S.D. Cova and P.D. Townsend (2005), *Quantum key distribution system clocked at 2 GHz*, Opt. Express, Vol. 13, pp. 3015-3020.
11. D. Lancho, J. Martinez, D. Elkouss, M. Soto and V. Martin (2009), *QKD in Standard Optical Telecommunications Networks*, in Proceedings of Int. ICST Conference on Quantum Communication and Quantum Networking (QuantumComm 2009), pp. 142-149, arXiv:1006.1858v1 [quant-ph].
12. P. Eraerds, N. Walenta, M. Legré, N. Gisin and H. Zbinden (2009), *Quantum key distribution and 1 Gbit/s data encryption over a single fibre*, arXiv:0912.1798v1 [quant-ph].
13. G. Van Assche (2006), *Quantum Cryptography and Secret-Key Distillation*, Cambridge University Press.
14. D. Slepian and J. Wolf (1973), *Noiseless coding of correlated information sources*, IEEE Trans. Inf. Theory, Vol. 19, No. 4, pp. 471-480.
15. R. Zamir, S. Shamai and U. Erez (2002), *Nested linear/lattice codes for structured multiterminal binning*, IEEE Trans. Inf. Theory, Vol. 48, No. 6, pp. 1250-1276.
16. C.H. Bennett, F. Bessette, G. Brassard, L. Salvail and J. Smolin (1992), *Experimental quantum cryptography*, J. Cryptology, Vol. 5, No. 1, pp. 3-28.
17. T.C. Ralph (1999), *Continuous variable quantum cryptography*, Phys. Rev. A, Vol. 61, No. 1, p. 010303.
18. F. Grosshans, P. Grangier and C.H. Bennett (2002), *Continuous Variable Quantum Cryptography Using Coherent States*, Phys. Rev. Lett., Vol. 88, No. 5, p. 057902.
19. S. Fossier, E. Diamanti, T. Debuisschert, R. Tualle-Brouri, and P. Grangier (2009), *Field test of a continuous-variable quantum key distribution prototype*, New. J. Phys., Vol. 11, No. 4, p. 045023.
20. D. Gottesman and H.-K. Lo (2003), *Proof of Security of Quantum Key Distribution With Two-Way Classical Communications*, IEEE Trans. Inf. Theory, Vol. 49, pp. 457-475.
21. A.D. Liveris, Zixiang Xiong and C.N. Georghiades (2002), *Compression of binary sources with side information at the decoder using LDPC codes*, IEEE Commun. Lett., Vol. 6, No. 10, pp. 440-442.
22. A.D. Wyner (1974), *Recent Results in the Shannon Theory*, IEEE Trans. Inf. Theory, Vol. 20, No. 1, pp. 2-10.
23. C. Elliott, A. Colvin, D. Pearson, O. Pikalo, J. Schlafer and H. Yeh (2005), *Current status of the DARPA Quantum Network*, quant-ph/0503058.
24. D. Elkouss, A. Leverrier, R. Alleaume and J.J. Boutros (2009), *Efficient reconciliation protocol for discrete-variable quantum key distribution*, IEEE Int. Symposium on Inf. Theory, pp. 1879-1883.
25. C.H. Bennett, G. Brassard, C. Crepeau and U.M. Maurer (1995), *Generalized privacy amplification*, IEEE Trans. Inf. Theory, Vol. 41, No. 6, pp. 1915-1923.
26. T.J. Richardson and R.L. Urbanke (2001), *The capacity of low-density parity-check codes under message-passing decoding*, IEEE Trans. Inf. Theory, Vol. 47, No. 2, pp. 599-618.
27. D. Elkouss, J. Martinez, D. Lancho and V. Martin (2010), *Rate Compatible Protocol for Information Reconciliation: An application to QKD*, IEEE Inf. Theory Workshop (ITW), pp. 145-149.
28. J. Ha, J. Kim and S.W. McLaughlin (2004), *Rate-compatible puncturing of low-density*

parity-check codes, IEEE Trans. Inf. Theory, Vol. 50, No. 11, pp. 2824-2836.
29. T. Tian and C.R. Jones (2005), *Construction of rate-compatible LDPC codes utilizing information shortening and parity puncturing*, EURASIP J. Wirel. Commun. Netw., Vol. 2005, No. 5, pp. 789-795.
30. M.A. Nielsen and I.L. Chuang (2000), *Quantum Computation and Quantum Information*, Cambridge University Press.
31. Z.L. Yuan, A.R. Dixon, J.F. Dynes, A.W. Sharpe and A.J. Shields (2008), *Practical gigahertz quantum key distribution based on avalanche photodiodes*, New J. Phys., Vol. 11, No. 4, p. 045019.
32. J. Martinez-Mateo, D. Elkouss and V. Martin (2010), *Improved construction of irregular progressive edge-growth Tanner graphs in the waterfall region*, IEEE Commun. Lett., accepted (to be published).
33. C. Crépeau (1995), *Réconcilliation et Distillation publiques de secret*, unpublished manuscript, available at http://www.cs.mcgill.ca/~crepeau/theses.html.